Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektronsko poslovanje

PHP OOP MYSQL

Domaći zadatak 1

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
| Branislav | Trajkov | 2019/0229 |
| Link ka Github-u | https://github.com/baneskiii/itehPrviDomaci.git | |

Sadržaj

[1 Korisnički zahtev 3](#_Toc118977244)

[2 Implementacija 8](#_Toc118977245)

[2.1 Kreiranje rezervacije 8](#_Toc118977246)

[2.2 Izmena rezervacije 10](#_Toc118977247)

[2.3 Brisanje rezervacije 11](#_Toc118977248)

[2.4 Pretraživanje rezervacije 13](#_Toc118977249)

[3 Korisničko uputstvo 16](#_Toc118977250)

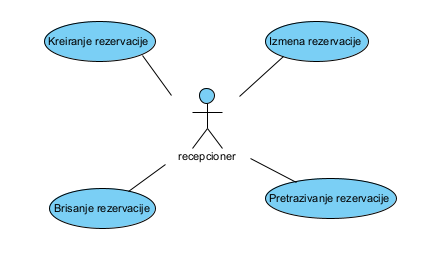
# Korisnički zahtev

Domaći se odnosi na izradu aplikacije koja će voditi evidenciju o radu hotelskog smeštaja. U okviru aplikacije vodiće se evidencija o gostima koji posećuju konkretan hotel, evidencija svih soba koje hotel ima na raspolaganju kao i evidencija ugovorenih rezervacija soba od strane gostiju koji planiraju da posete i koriste usluge hotelskog smeštaja.

Sama aplikacija omogućava funkcionisanje na sledeći način. Dakle, aplikacija omogućava da recepcioner, prilikom boravka novih gostiju, te goste u aplikaciji evidentira i njihove podatke unese u sistem. Ukoliko postoji potreba za time, recepcioner ima mogućnost da podatke rezervacije, u slučaju greške, izmeni i te izmene sačuva u sistem. Za potrebe sticanja uvida o rezervacijama, recepcioner može vršiti pretraživanje njihovih podataka. U slučaju da određeni gost, u trenutku provere, više ne boravi u hotelu, moguće je brisanje njegove rezervacije od strane recepcionera. Takođe, recepcioner ima mogućnost da sagleda podatke o svim sobama koje postoje u okviru hotela.

U ovoj aplikaciji identifikovano je četiri slučajeva korišćenja:

1. Kreiranje rezervacije;
2. Izmena rezervacije;
3. Brisanje rezervacije;
4. Pretraživanje rezervacije.



SK1: Slučaj korišćenja – Kreiranje rezervacije

Naziv SK

Kreiranje rezervacije

Aktori SK

Recepcioner

Učesnici SK

Recepcioner i sistem (program)

Preduslov: Sistem je uključen i recepcioner je prijavljen pod svojom šifrom. Sistem prikazuje formu za rad sa rezervacijom.

Osnovni scenario SK

1. Recepcioner unosi podatke o rezervaciji. (APUSO)
2. Recepcioner kontroliše da li je korektno uneo podatke o rezervaciji. (ANSO)
3. Recepcioner poziva sistem da zapamti podatke o rezervaciji. (APSO)
4. Sistem pamti podatke o rezervaciji. (SO)
5. Sistem prikazuje recepcioneru zapamćenu rezervaciju i poruku: „Rezervacija je dodata“. (IA)

Alternativna scenarija

5.1 Ukoliko sistem ne može da zapamti podatke o rezervaciji on prikazuje recepcioneru poruku: „Rezervacija nije dodata”. (IA)

SK2: Slučaj korišćenja – Izmena rezervacije

Naziv SK

Izmena rezervacije

Aktori SK

Recepcioner

Učesnici SK

Recepcioner i sistem (program)

Preduslov: Sistem je uključen i recepcioner je prijavljen pod svojom šifrom. Sistem prikazuje formu za rad sa rezervacijom.

Osnovni scenario SK

1. Recepcioner unosi (menja) podatke o rezervaciji. (APUSO)
2. Recepcioner kontroliše da li je ispravno uneo podatke o rezervaciji. (ANSO)
3. Recepcioner poziva sistem da zapamti podatke o rezervaciji. (APSO)
4. Sistem pamti podatke o rezervaciji. (SO)
5. Sistem prikazuje recepcioneru zapamćenu rezervaciju i poruku: „Rezervacija je promenjena“. (IA)

Alternativna scenarija

* 1. Ukoliko sistem ne može da zapamti podatke o rezervaciji on prikazuje recepcioneru poruku: „Rezervacija nije promenjena“. (IA)

SK3: Slučaj korišćenja – Brisanje rezervacije

Naziv SK

Brisanje rezervacije

Aktori SK

Recepcioner

Učesnici SK

Recepcioner i sistem (program)

Preduslov: Sistem je uključen i recepcioner je prijavljen pod svojom šifrom. Sistem prikazuje formu za rad sa rezervacijom.

Osnovni scenario SK

1. Recepcioner bira rezervaciju. (APUSO)
2. Recepcioner poziva sistem da obriše podatke o rezervaciji. (APSO)
3. Sistem briše podatke o rezervaciji. (SO)
4. Sistem prikazuje recepcioneru poruku: “Rezervacija je obrisana.” (IA)

Alternativna scenarija

* 1. Ukoliko sistem ne može da obriše podatke o rezervaciji on prikazuje recepcioneru poruku: „Rezervacija nije obrisana”. (IA)

SK4: Slučaj korišćenja – Pretraživanje rezervacije

Naziv SK

Pretraživanje rezervacije

Aktori SK

Recepcioner

Učesnici SK

Recepcioner i sistem (program)

Preduslov: Sistem je uključen i recepcioner je prijavljen pod svojom šifrom. Sistem prikazuje formu za rad sa rezervacijom.

Osnovni scenario SK

1. Recepcioner unosi vrednost po kojoj pretražuje rezervacije. (APUSO)
2. Recepcioner poziva sistem da nađe rezervacije po zadatoj vrednosti. (APSO)
3. Sistem traži rezervacije po zadatoj vrednosti. (SO)
4. Sistem prikazuje recepcioneru podatke o rezervacijama. (IA)

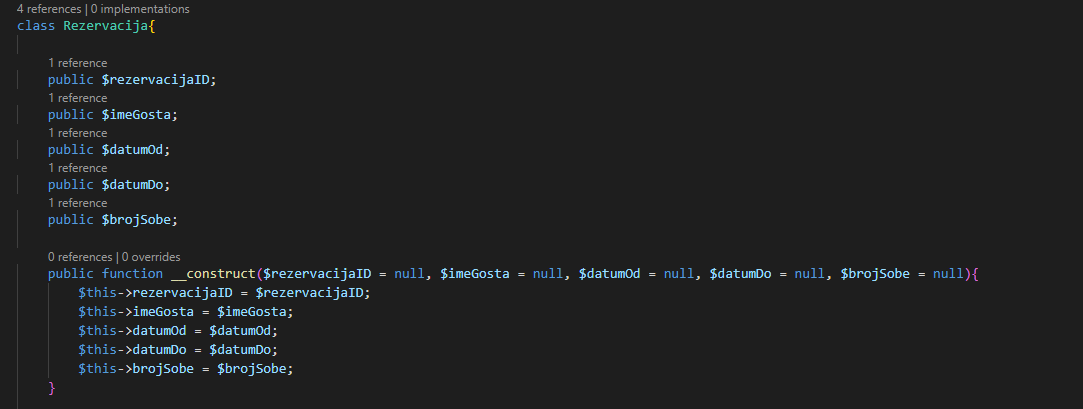
Alternativna scenarija

4.1. Ukoliko sistem ne može da nađe rezervacije on recepcioneru prikazuje prazan spisak rezervacija. (IA)

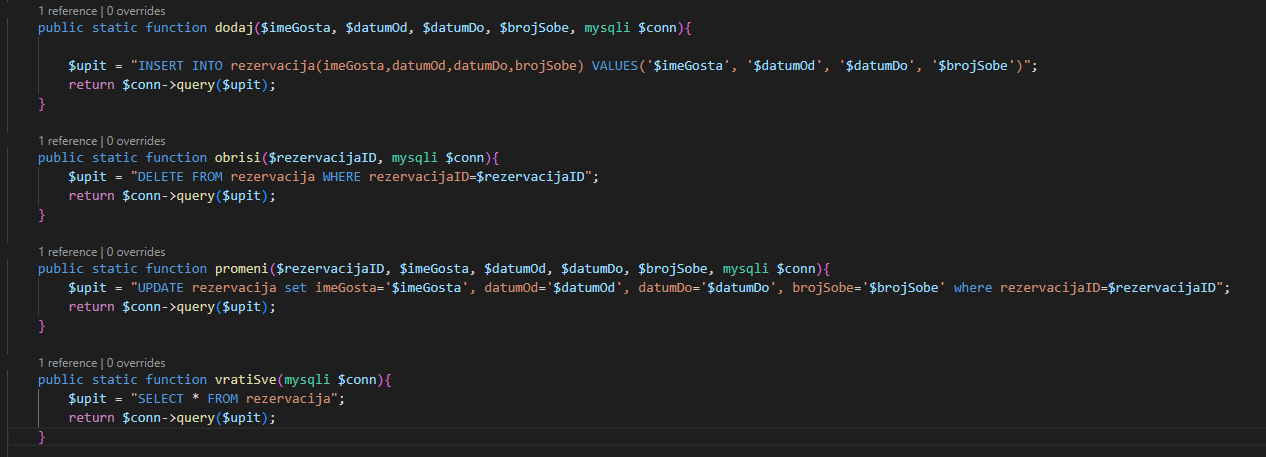
# Implementacija

## Kreiranje rezervacije

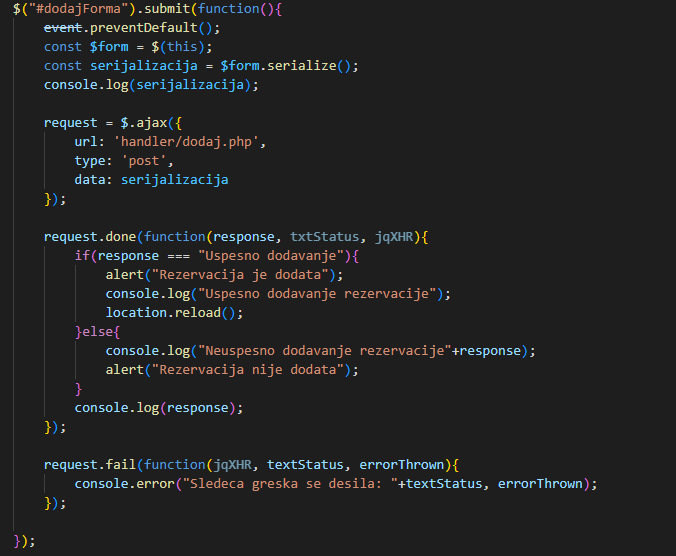
U aplikaciji, napravljena je klasa Rezervacija koja izgleda ovako:



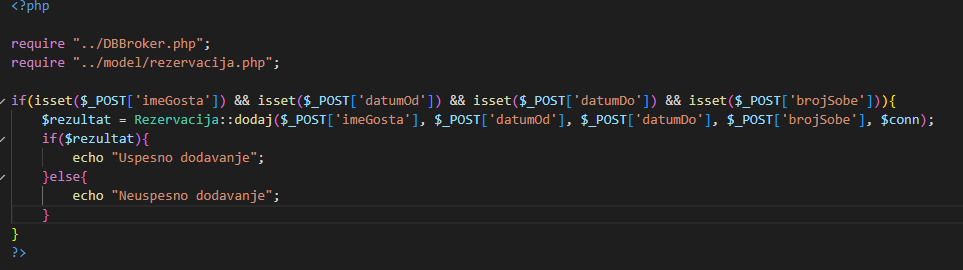
Kao što se može primetiti, ova klasa ima atribute rezervacijaID koji predstavlja primarni ključ ove klase, imeGosta, datumOd, datumDo, brojSobe koji je spoljašnji ključ prema tabeli Soba u bazi podataka i konekciju. Pored ovih atribura i konstruktora, klasa ima i sledeće funkcije:



Funkcija dodaj kao svoje ulazne argumente ima ime gosta, datum od, datum do i broj sobe, kao podatke koje je potrebno uneti u bazu. Kreira se sql insert upit koji se izvršava, i funkcija vraća rezultat izvršenja sql upita za dodavanje rezervacije. Kada se klikne na dugme dodaj u okviru dodaj forme, izvršava se sledeća funkcija koja se nalazi u main.js fajlu:



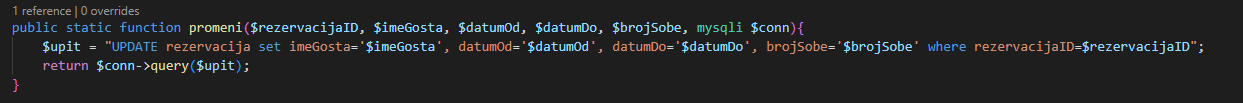
Prilikom izvršavanja funkcije, korišćenjem AJAX-a, prihvataju se podaci, preko POST zahteva, koje je recepcioner uneo u formi. Ukoliko je izvršenje uspešno, na ekranu se pojavljuje poruka da je rezervacija uspešno dodata, u suprotnom, prikazuje se poruka da rezervacija nije uspešno dodata u bazi. Ovde se takođe koristi još i fajl dodaj.php koji izgleda ovako:



Fajl uključuje DBBroker.php i klasu Rezervacija kako bi se mogla izvršiti funkcija dodaj. Proverava se da li POST globalna promenljiva ima neophodne podatke za izvršenje funkcije dodaj, i ukoliko je to slučaj, funkcija se izvršava i njen rezultat se čuva u promenljivoj rezultat, na osnovu koje funkcija u main.js fajlu recepcioneru na ekanu ispisuje obaveštenje o dodavanju rezervacije.

## Izmena rezervacije

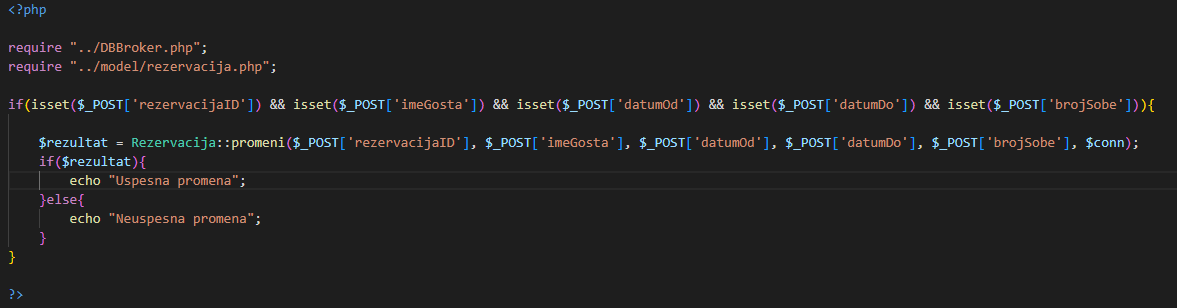
U klasi Rezervacija nalazi se funkcija promeni koja izgleda ovako:



Funkcija promeni ima sledeće ulazne argumente: rezervacijaID, ime gosta, datum od, datum do, broj sobe i konekciju. Promenljiva upit predsavlja sql update upit koji prihvata ove ulazne argumente kako bi se rezervacija na osnovu njih izmenila. Sql upit se izvršava i njegov rezultat predstavlja povratnu vrednost funkcije promeni. Slično kao i u slučaju dodavanja, u main.js fajlu se nalazi sledeća funkcija:



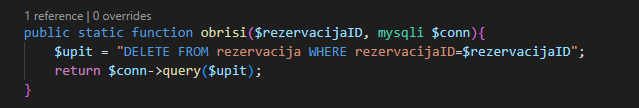
Funkcija funkcioniše slično kao i prilikom dodavanja rezervacije. Dakle, ona prima podatke koji su uneti u formi za izmenu podataka rezervacije, pravi se AJAX request i poziva se fajl promeni.php:



Ovaj fajl uključuje DBBroker.php i klasu Rezervacija, kako bi se izvršila funkcija promeni. Provera va se da li POST globalna promenljiva ima sve neophodne parametre za izvršenje ove funkcije, i ako je to slučaj, ona se izvršava i njen rezultat se čuva u promenljivoj rezultat. Na osnovu ovog rezultata, recepcioneru se prikazuje poruka o uspešnosti promene.

## Brisanje rezervacije

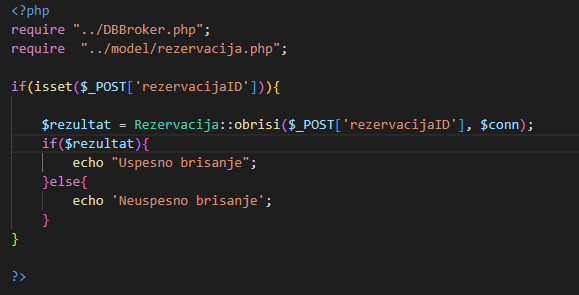
Klasa Rezervacija u okviru sebe sadrži funkciju obriši:



Ova funkcija, za razliku od prethodne dve koje se nalaze u klasi Rezervacija, kao svoj ulazni argument prihvata samo parametar rezervacijaID i konekciju, koji predstavlja ID one rezervacije koju je potrebno obrisati iz baze. Slično kao i do sada, u funkciji se kreira sql delete upit koji prihvata ID rezervacije koju je potrebno obrisati. Sql upit se realizuje i vraća se njegov rezultat kao povratna vrednost funkcije obrisi. U main.js fajlu imamo sledeću funkciju:



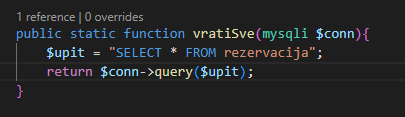
Ona prima podatke sa obriši forme u koju recepcioner unosi podatke na osnovu kojih se rezervacija briše, prosleđuje ih do AJAX requesta koji poziva obrisi.php fajl koji se nalazi u folderu handler:



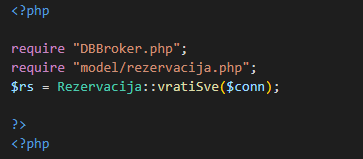
Fajl obrisi.php uključuje DBBroker.php i klasu Rezervacija. Nakon toga se proverava da li POST globalna promenljiva sadrži ID rezervacije koju je potrebno obrisati, i ako sadrži poziva se funkcija obrisi nad klasom Rezervacija. Kada se izvrši, njena povratna vrednost se smešta u promenljivu rezultat, na osnovu koje se prikazuje poruka recepcioneru o uspešnosti brisanja rezervacije.

## Pretraživanje rezervacije

Pored do sada navedenih funkcija, klasa Rezervacija sadrži i funkciju vratiSve:



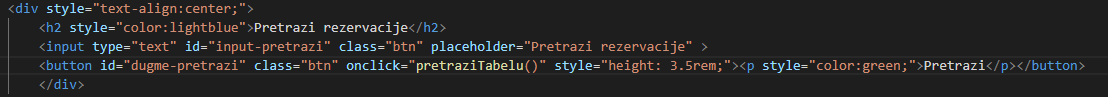
Ova funkcija kao svoj ulazni parmetar prima samo konekciju sa bazom podataka. Kreira se sql select upit koji treba da vrati sve podatke rezervacija koje se nalaze u bazi. Upit se izvršava i funkcija vraća povratnu vrednost izvršenja ovog sql upita.



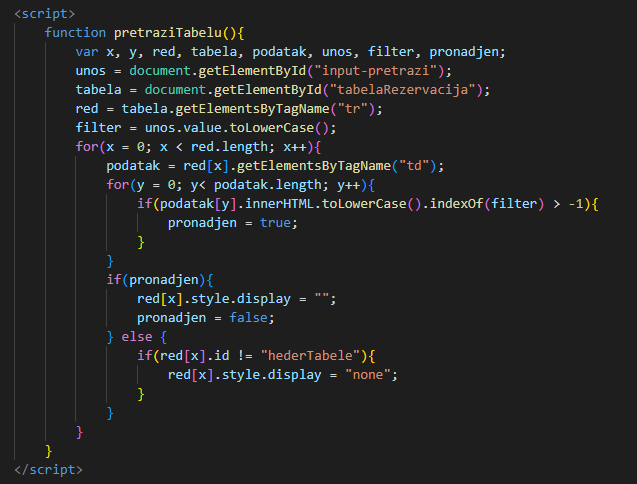
Na samom početku fajla reservation.php, uključuju se DBBroker.php i klasa Rezervacija kao što je rađeno i ranije, kako bi se mogla pozvati funkcija vratiSve koja se ne nalazi u ovom fajlu. Funkcija vratiSve se poziva i njen rezultat se čuva u promenljivoj rs.



Promenljiva rs zapravi sadrži rezultat sql select upita, što je u ovom slučaju cela tabela rezervacija iz baze. Kako bi se ti podaci prikazali u tabeli, koristimo while petlju i funkciju fetch\_array(). Funkcija fetch\_array() vraća red po red tabele Rezervacija i smešta ga u promenljivu red, a onda se korišćenjem echo funkcije ispisuju vrednosti atributa rezervacije u okviru <td> taga tabele rezervacija. Postupak se ponavlja sve dok se ne prođe kroz sve redove tabele rezervacija.



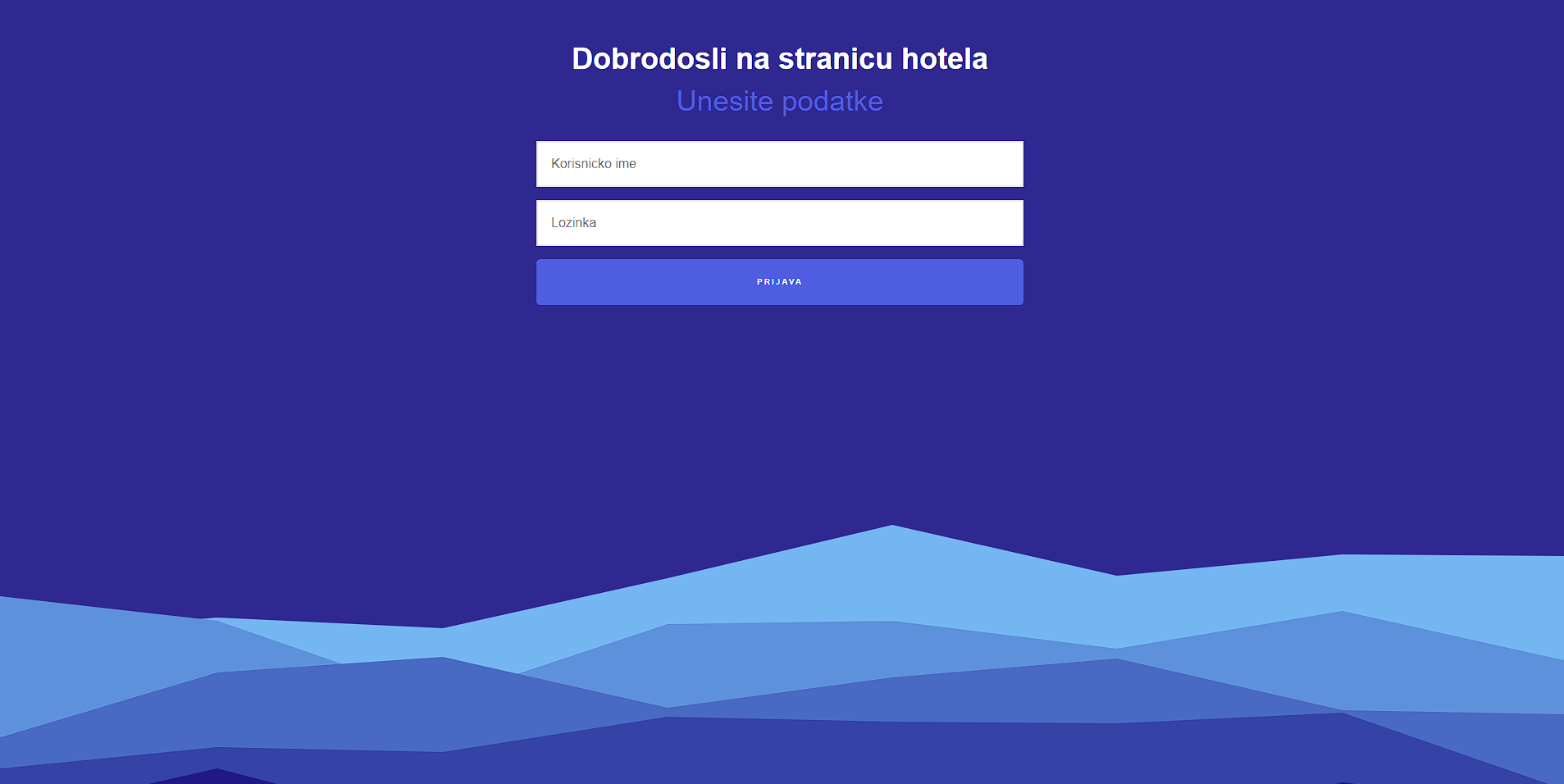
Na stranici se takođe nalazi input polje koje prima vrednost po kojoj recepcioner želi da pretražuje rezervacije, kao i dugme pretraži. Klikom na dugme pretraži, poziva se funkcija pretraziTabelu().



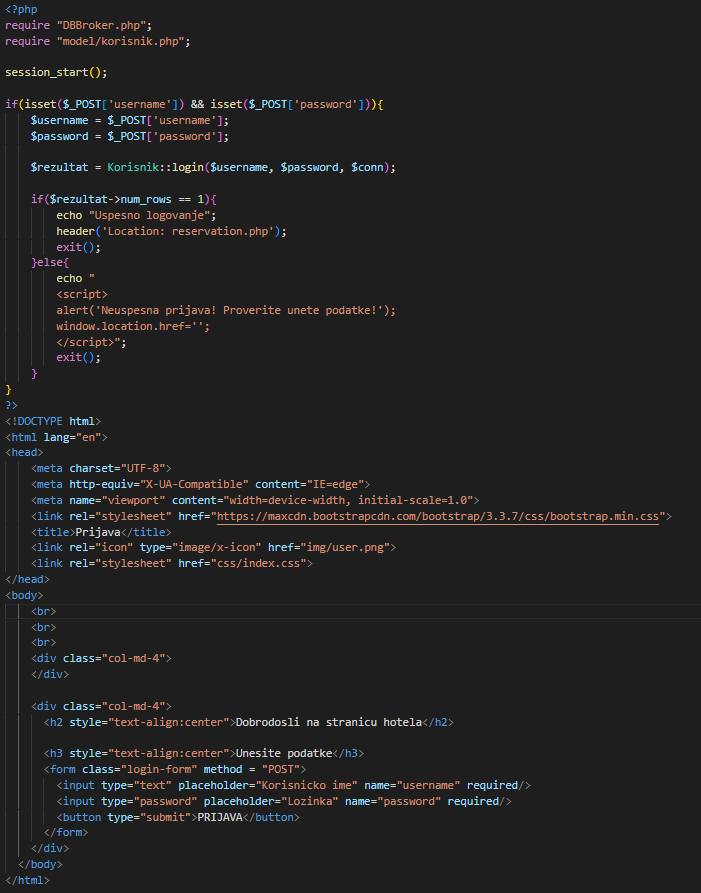
Funkcija pretraziTabelu() uzima unos recepcionera i tabelu sa podacima o svim rezervacijama i prolazi kroz svaki red i svaku kolonu tabele i proverava da li se unos recepcionera nalazi u nekoj ćeliji tabele. Ako je to slučaj, taj red tabele se prikazuje. Ako to nije slučaj, taj red se zanemaruje, preskače i ne prikazuje. Takođe, pretraga je case insensitive, odnosno ne pravi razliku između malih i velikih slova kada proverava sam unos recepcionera.

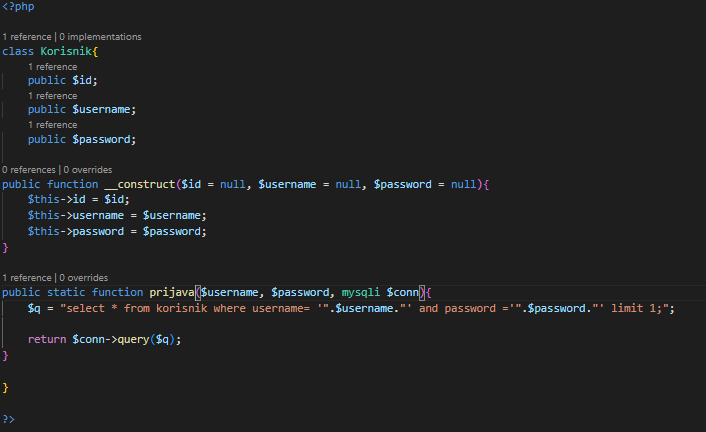
# Korisničko uputstvo

Prilikom korišćenja aplikacije, prvo na šta recepcioner nailazi jeste stranica za prijavu na aplikaciju.

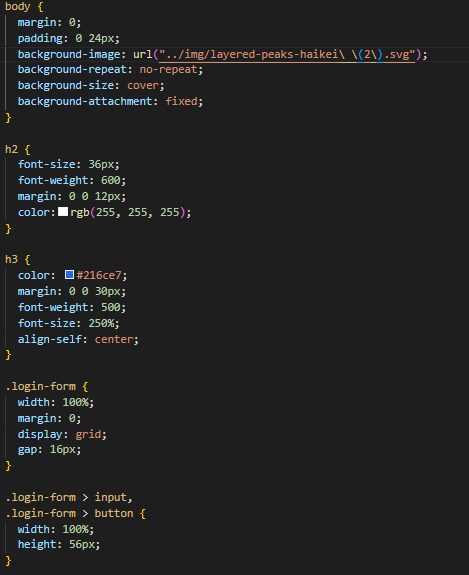


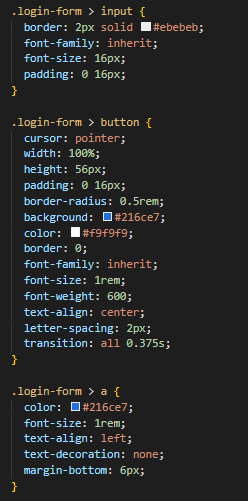
Od recepcionera se traži da se prijavi unošenjem korisničkog imena i lozinke. U bazi se čuvaju podaci za prijavu svih recepcionera koji imaju dozvoljen pristup za korišćenje aplikacije, tako da ukoliko se podaci koje recepcioner unese ne poklope sa tim podacima, ispisuje se poruka da je prijava neuspešna i recepcioner mora ponovo da unese podatke. U kodu to izgleda ovako:





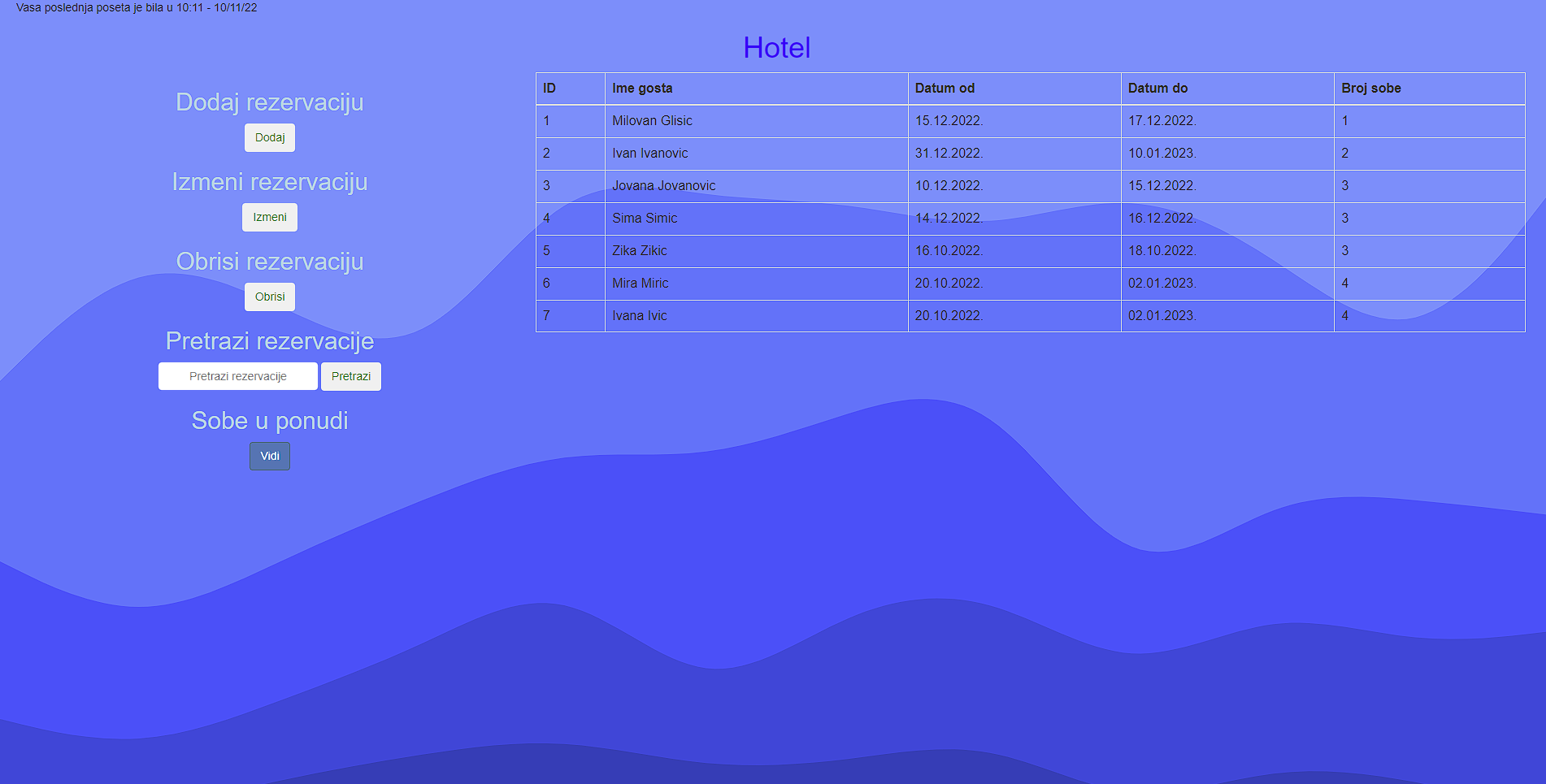
Podaci koje recepcioner unese prilikom prijave se čuvaju u POST globalnoj promenljivoj. Ukoliko oni postoje poziva se funkcija prijava nad klasom Korisnik koja vraća recepcionera iz baze koji sadrži te podatke. Ukoliko on postoji, prijava je uspešna i recepcioneru se prikazuje stranica reservation.php, u suprotnom on se mora ponovo prijaviti.



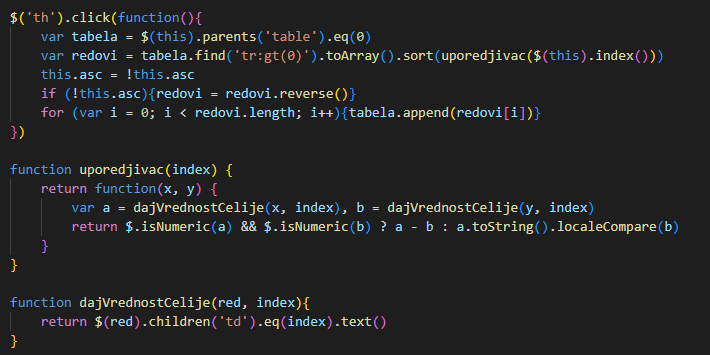


U fajlu index.css se nalaze sva css podešavanja vezana za izgled komponenata stranice index.php, odnoso stranice za prijavu recepcionera. U folderu img nalaze se sve slike koje su korišćene na svim stranicama aplikacije.

Ukoliko je prijava recepcionera bila uspešna, njemu se prikazuje sledeća stranica:

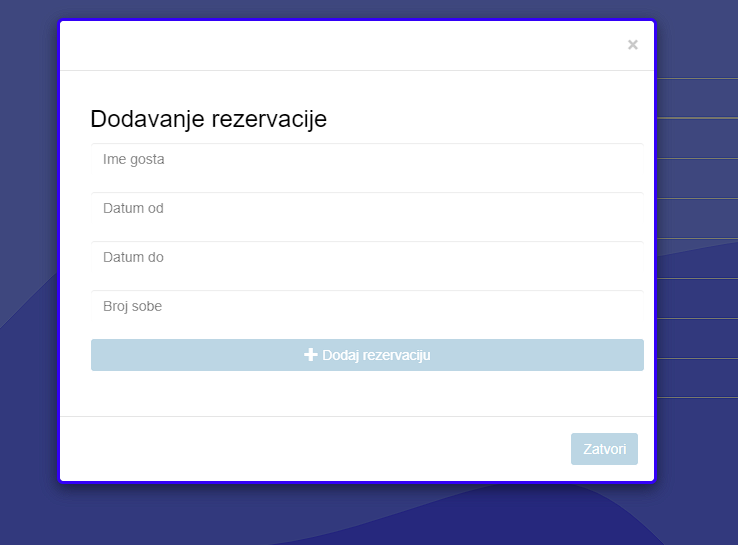


Pre svega, u tabeli sa desne strane, prikazuju se sve rezervacije koje se trenutno nalaze u tabeli rezervacija baze podataka hotel. Klikom na heder bilo koje kolone tabele, vrši se sortiranje rastuće i opadajuće podataka u tabeli.



Sortiranje je omogućeno ovim funkcijama koje se nalaze u fajlu main.js. Klikom da bilo koji heder tabele, funckija uzima tabelu i njene redove i sortira ih. Ako su redovi bili sortirani rastuće novim klikom biće sorturani opadajuće, i obrnuto, i nakon toga se redovi dodaju u postijeću tabelu. Tu imamo još i pomoćne funkcije upoređivač, koja omogućava poređenje stringova i brojeva, kao i funkcija dajVrednostCelije, koja za prosleđeni red i indeks, vraća tekst ćelije iz tabele iz tog reda i sa tim indeksom.

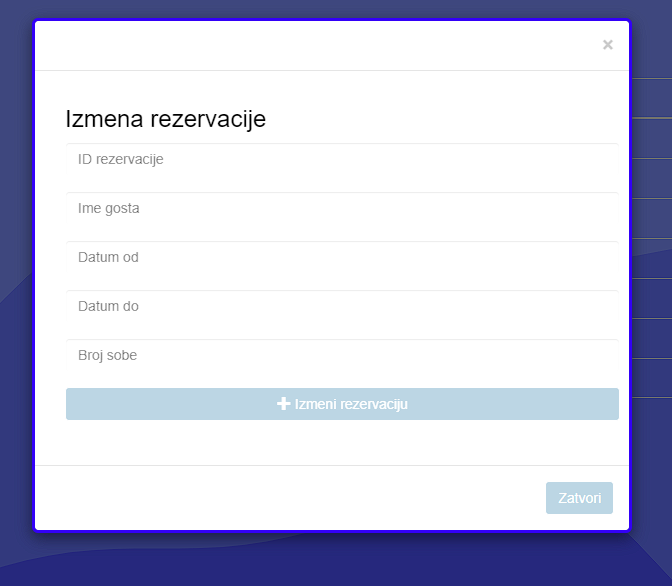
Klikom na dugme Dodaj otvara se modal za unošenje podataka rezervacije koju recepcioner želi da unese u bazu:

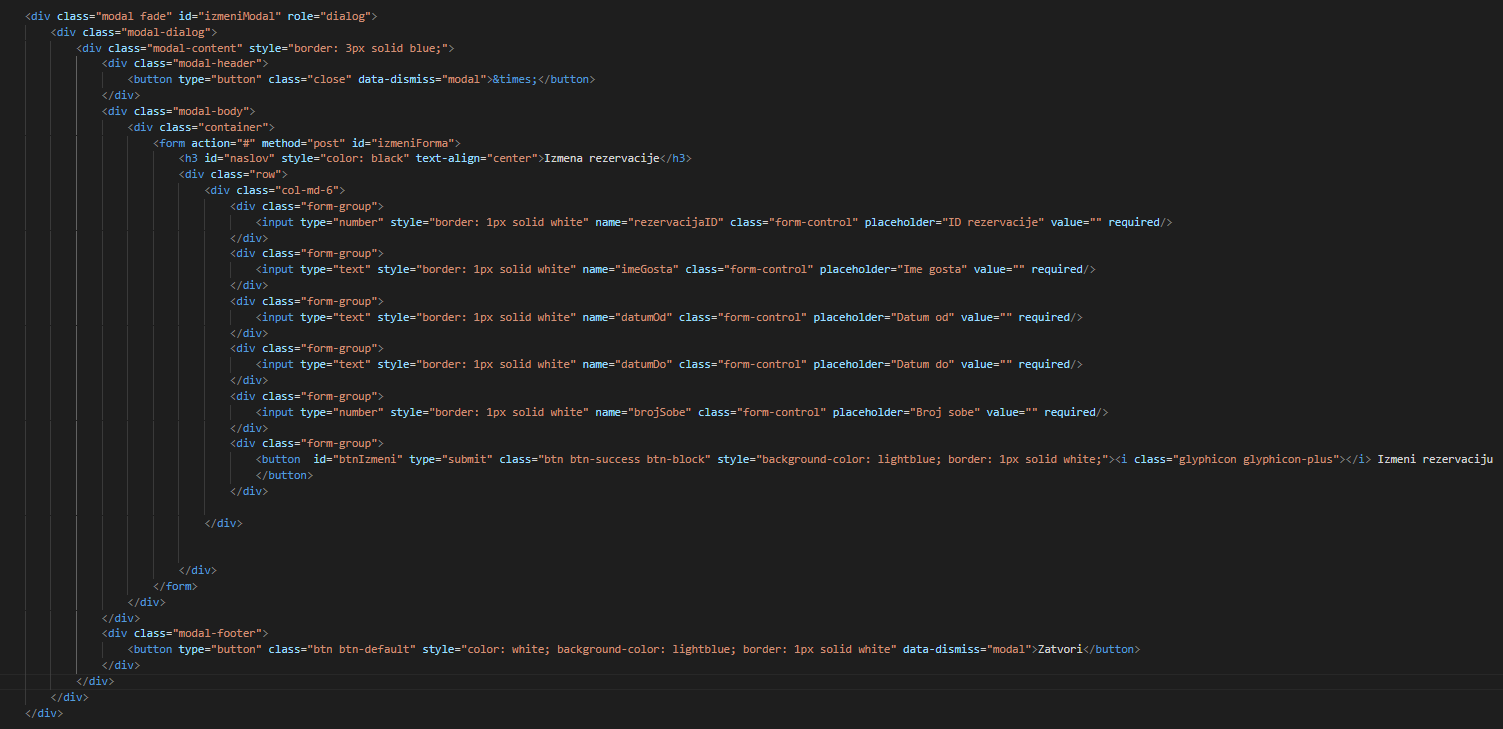




Modal sadrži input polja za unos imena gosta, datuma od, datuma do i broja sobe u kojoj će gost boraviti, kao i dva dugmeta, jedno za dodavanje rezervacije i jedno za zatvaranje modala.

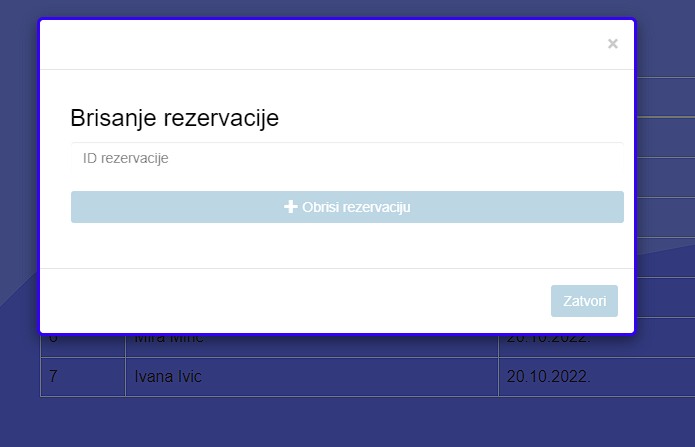
Klikom na dugme Izmeni otvara se modal za izmenu podataka već postojeće rezervacije u bazi:





Modal, pored polja za unos imena gosta, datuma od, datuma do i broja sobe, sadrži i polje za unos ID-ja rezervacije čiji će podaci biti promenjeni u bazi podataka. Pored toga, i ova forma sadrži dugme za izmenu rezervacije i dugme za zatvaranje modala.

Klikom na dugme Obrisi otvara se modal za brisanje rezervacije iz baze:

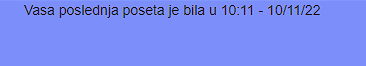


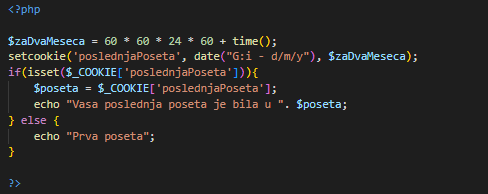
Ovaj modal je veoma jednostavan, sadrži samo polje za unos ID-ja rezervacije koju recepcioner želi da obriše, dugme za brisanje i dugme za zatvaranje samog modala.

Što se tiče pretraživanja rezervacija, u polje za unos, recepcioner upisuje vrednost koju traži u tabeli i klikom na dugme Pretrazi, pretraga se izvršava na već opisan način, i u tabeli se prikazuju podaci koje je recepcioner tražio:



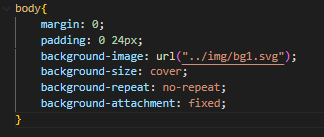
Takođe, u gornjem levom uglu se, korišćenjem kolačića, ispisuje poruka, „Prva poseta“, ako recepcioner prvu put ulazi na sajt, ili se ispisuje vreme i datum kada je poslednji put pristupio sajtu:





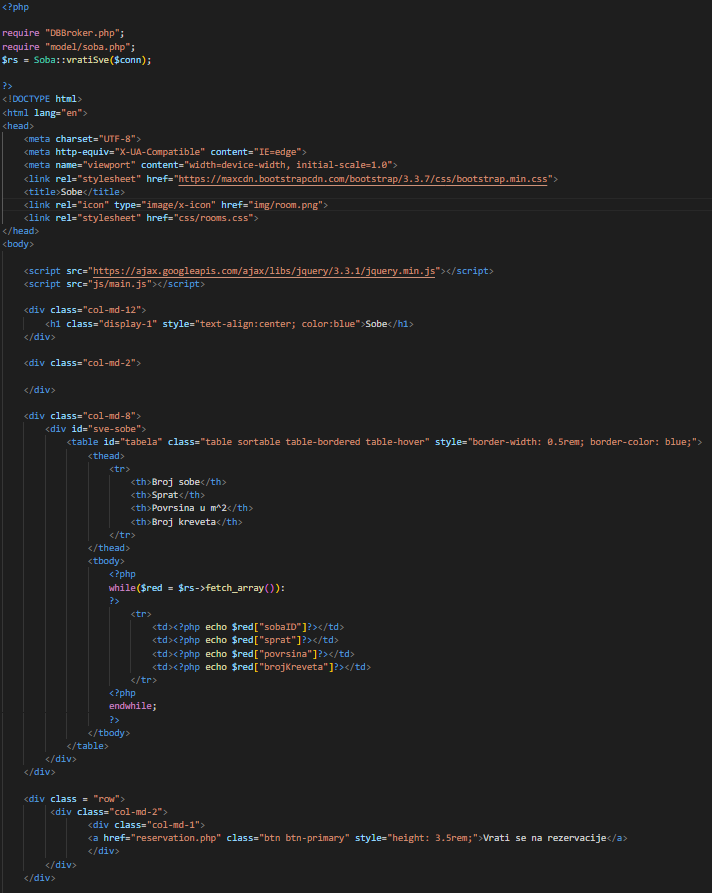
Na samom početku stranice reservation.php se postavlja kolačić koji traje 2 meseca i koji čuva podatak o poslednjoj poseti stranici. Ako kolačić do sada nije postojao, on se kreira i ispisuje se poruka o prvoj poseti. U suprotnom, ispisuje se poruka kada je recepcioner poslednji put posetio stranicu.

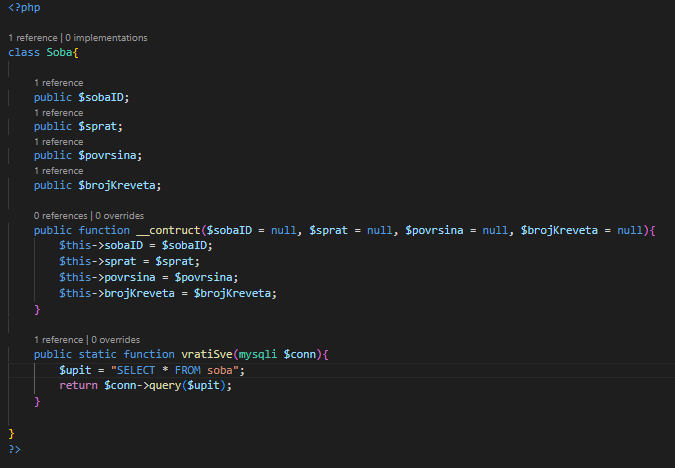
U fajlu reservation.css se nalate css postavke vezane za izgled stranice:



Klikom da dugme Vidi, recepcioneru se prikazuje stranica rooms.php:

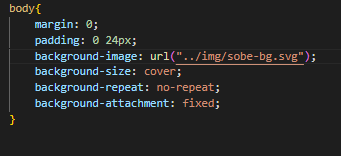




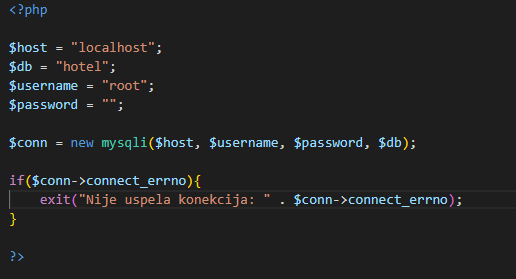


Slično kao i kod rezervacija, na ovoj stranici se nalazi tabela u kojoj se prikazuju svi podaci o sobama koje postoje u hotelu. Ovaj prikaz je urađen na sličan način kao i prikaz rezervacija, s obzirom da i klasa Soba ima funkciju vratiSve, koja vraća sve sobe iz tabele soba baze hotel. Prolaskom kroz while petlju, podaci o sobama se ispisuju u tabeli. Pored toga, i u ovoj tabeli je omogućeno sortiranje po kolonama. Takođe, ako recepcioner klikne na dugme Vrati se na rezervacije, biće prebačen nazad na stranicu sa rezervacijama.

U fajlu rooms.css se nalaze css postavke vezane za stranicu rooms.php:



Na kraju, potrebno je navesti da fajl DBBroker.php predstavlja broker baze podataka:



U ovom fajlu se, korišćenjem podataka o hostu, nazivu baze, username-u i password-u za pristup ovoj bazi, kreira konekcija conn sa bazom podataka hotel. Ukoliko ova konekcija iz nekog razloga nije uspela, prikazuje se poruka da samo uspostavljanje konekcije nije uspelo.